**PROJETO INTERDISCIPLINAR**

**Criação De Um Banco Digital**

**MidoBank**

**São Paulo**

**UNIVERSIDADE CRUZEIRO DO SUL**

**Projeto Interdisciplinar – 2023 – 1º semestre**

**Engenharia de Software**

**Análise e Projeto de Sistemas I**

**Curso:** CST em Análise e Desenvolvimento de Sistemas (ADS)

**Semestres:** 2º B Noturno **Campus:** SM – São Miguel  
  
**Disciplinas Envolvidas:**

**1- Análise e Projetos de Sistemas I:** Como analista de sistemas, é imprescindível que sejam abstraídos cenários propostos.  
**2- Engenharia de Software:** Elaboração de toda a documentação de Engenharia de Software, incluindo requisitos funcionais, não funcionais, regras de negócio, diagrama de classes, sequência e demais itens que julgar necessário.  
  
**Nomes:**   
Anderson Franco Ribeiro Júnior, 31033849;  
Igor Barcelo, 31099858  
Guilherme Miranda da Silva, 30950732;  
Natan Carvalho da Silva, 31299440;  
Victor Frohlich, 31187382

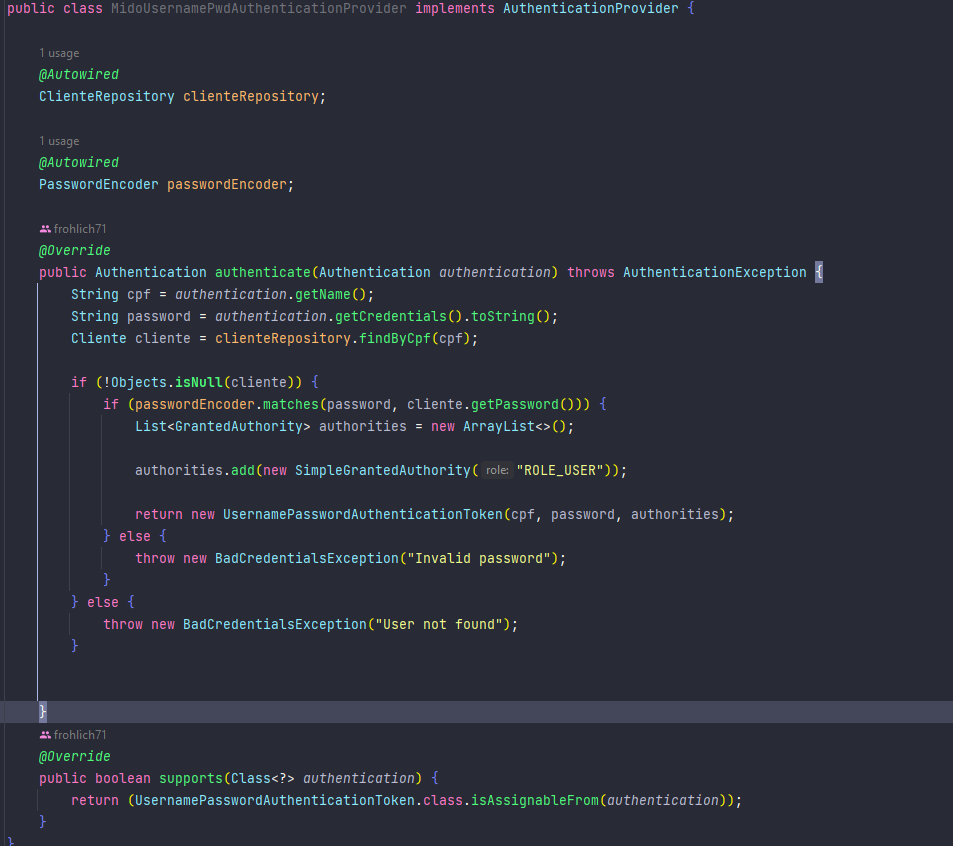
**Requisitos Funcionais**

* **Autenticação**

° O sistema deve permitir que o usuário faça login utilizando seu CPF e senha;

° Após a autenticação, o sistema deve gerar um token de sessão válido por um período determinado de tempo;

° O sistema deve permitir que o usuário realize ações somente após a autenticação e validação do token de sessão.

* **Classe criada para autenticação e tokenização:**  
  

* **Transações**

**PIX**

O sistema deve permitir que o usuário realize transações via PIX, informando o valor a ser transferido e os dados do destinatário (chave PIX, CPF ou conta bancária), caso haja saldo suficiente;



**TED**

O sistema deve permitir que o usuário realize transferências de banco para banco, informando o valor a ser transferido e os dados do destinatário (nome do banco, número da agência, número da conta e CPF), caso haja saldo suficiente;



**SAQUE**

O sistema deve permitir que o usuário realize saques em caixas eletrônicos, informando o valor a ser sacado e validando a autenticação do usuário, caso haja saldo suficiente;



**DEPÓSITO**

O sistema deve permitir que o usuário realize depósitos em caixas eletrônicos, informando o valor a ser depositado e validando a autenticação do usuário, caso haja saldo suficiente.

**Consultas**

° O sistema deve permitir que o usuário consulte o saldo de sua conta bancária;

**°** O sistema deve permitir que o usuário consulte o histórico de transações realizadas em sua conta bancária;

° O sistema deve permitir que o usuário consulte os dados de sua conta bancária (nome, CPF, número da agência, número da conta).

**Requisitos não-funcionais**

**Segurança**

° O sistema deve ser protegido contra ataques cibernéticos, utilizando técnicas de criptografia e protocolos de segurança;

° O sistema deve permitir que o usuário defina uma senha forte;

° O sistema deve permitir que o usuário recupere sua senha em caso de esquecimento, por meio de um processo seguro de redefinição de senha.

* **Classe com os métodos de segurança necessários para impedir invasões e garantir o sigilo de informações:**



**Usabilidade**

° O sistema deve ser intuitivo e fácil de usar, com uma interface amigável e clara;

° O sistema deve ser capaz de lidar com um grande volume de transações simultâneas, sem comprometer a performance ou a segurança.

**Performance**

° O sistema deve ser rápido e responsivo, permitindo que o usuário realize transações e consultas em tempo real;

° O sistema deve ser capaz de lidar com um grande volume de transações simultâneas, sem comprometer a performance ou a segurança.

**Disponibilidade**

° O sistema deve estar disponível 24 horas por dia, 7 dias por semana, com uma disponibilidade mínima de 99%;

° O sistema deve ser capaz de se recuperar rapidamente de falhas e interrupções, minimizando o impacto para o usuário.

**Manutenção**

° O sistema deve ser fácil de manter e atualizar, permitindo que novas funcionalidades sejam adicionadas com facilidade;

**Regras de Negócio**

° O valor máximo permitido para transações via PIX é de R$10.000 por dia, por usuário;

° É obrigatório informar a chave PIX, CPF ou conta bancária do destinatário para realizar transações via PIX ou transferências de banco para banco;

° O sistema deve gerar comprovantes de transação para o usuário, contendo informações como o valor transferido, o destinatário, a data e hora da transação;

° O sistema deve validar as informações de conta bancária do destinatário antes de permitir transferências.

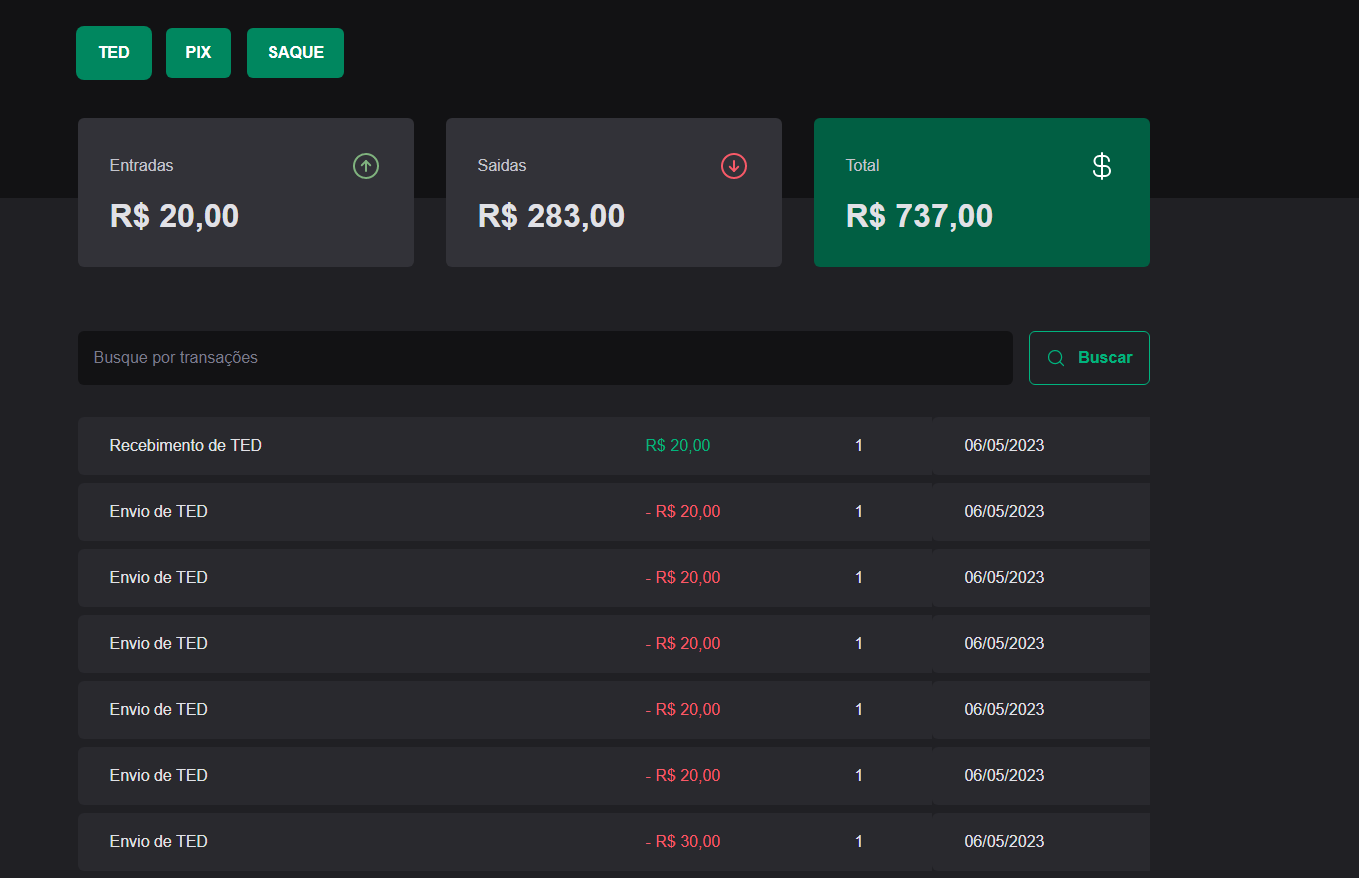
**MIDO BANK – STARTUP DE BANCO DIGITAL**

Startup de banco sem agência física voltado à transações digitais contendo as seguintes funcionalidades:

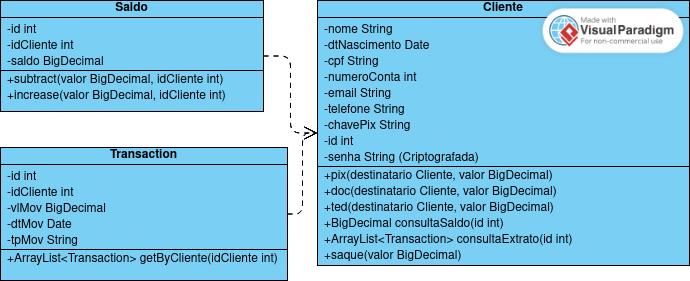
**Cadastros de clientes e chaves PIX;**

**Transações - PIX E TED; Saldo; Extrato; Depósito e Saque;**

**DASHBOARD – Tela inicial do usuário**

Todas as funcionalidades para usuários clientes estão disponibilizadas no dashboard de início do usuário;

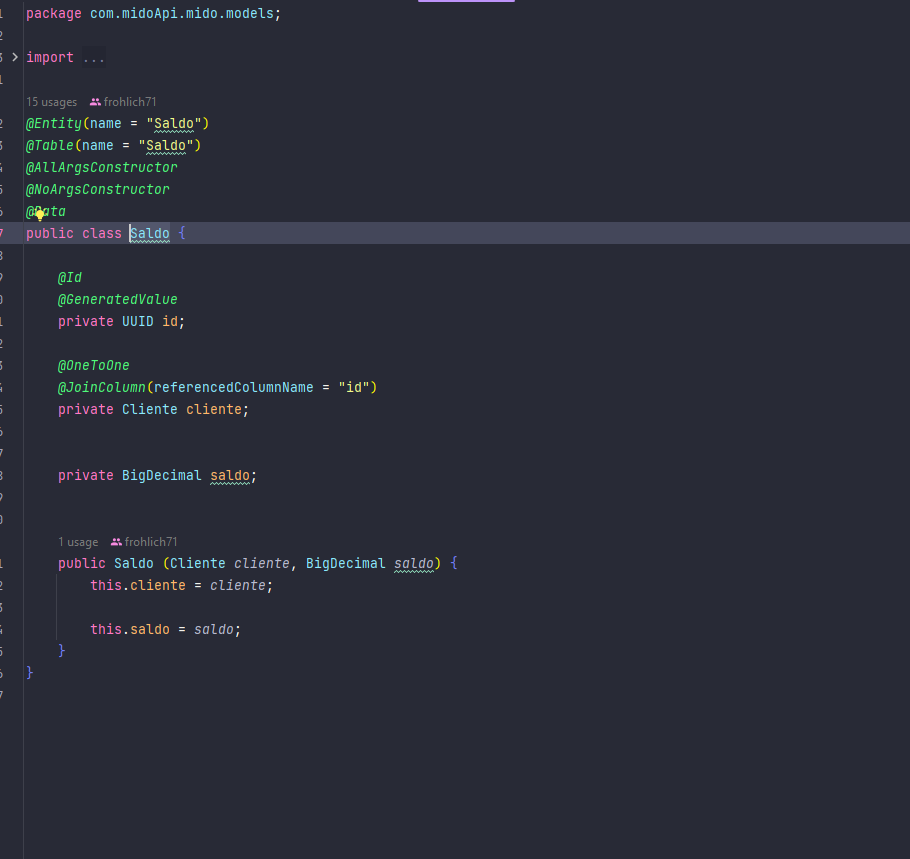
**Diagrama UML:**

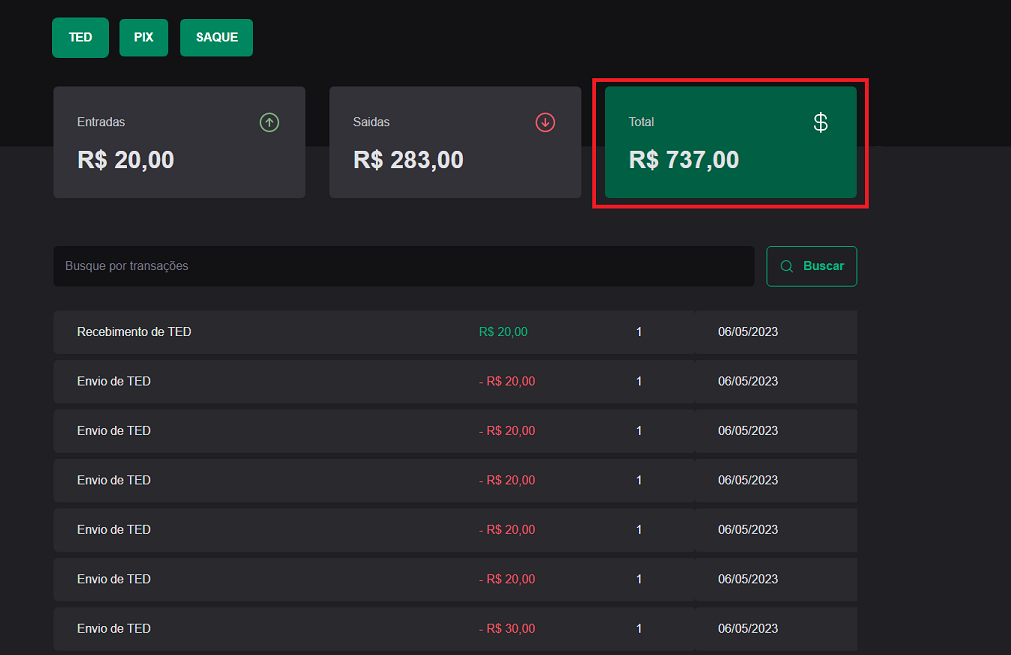


**Funcionalidades:**

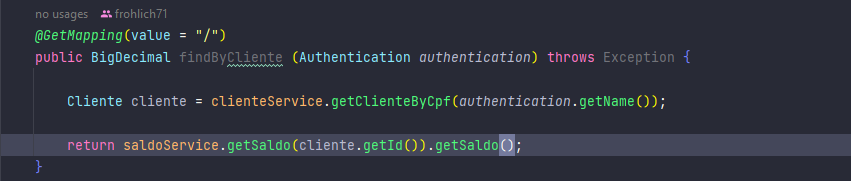
**Saldo -**

* + **Classe Saldo no sistema:**



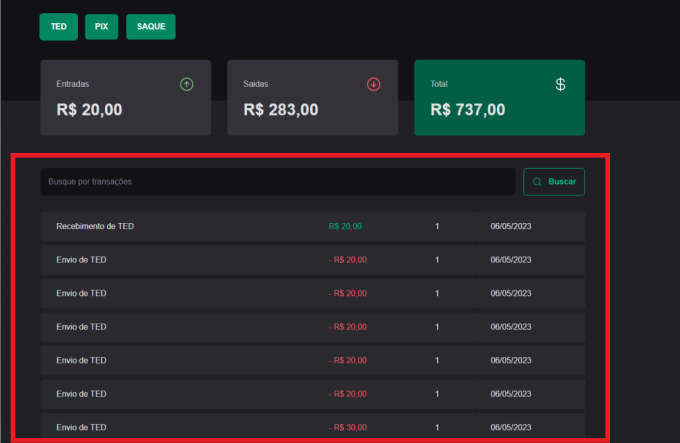


* O saldo é alterado conforme é realizado transferências na conta bancária do usuário, sinalizando atualizações do saldo conforme transferências realizadas pelo usuário;



* + Função de exibição de saldo

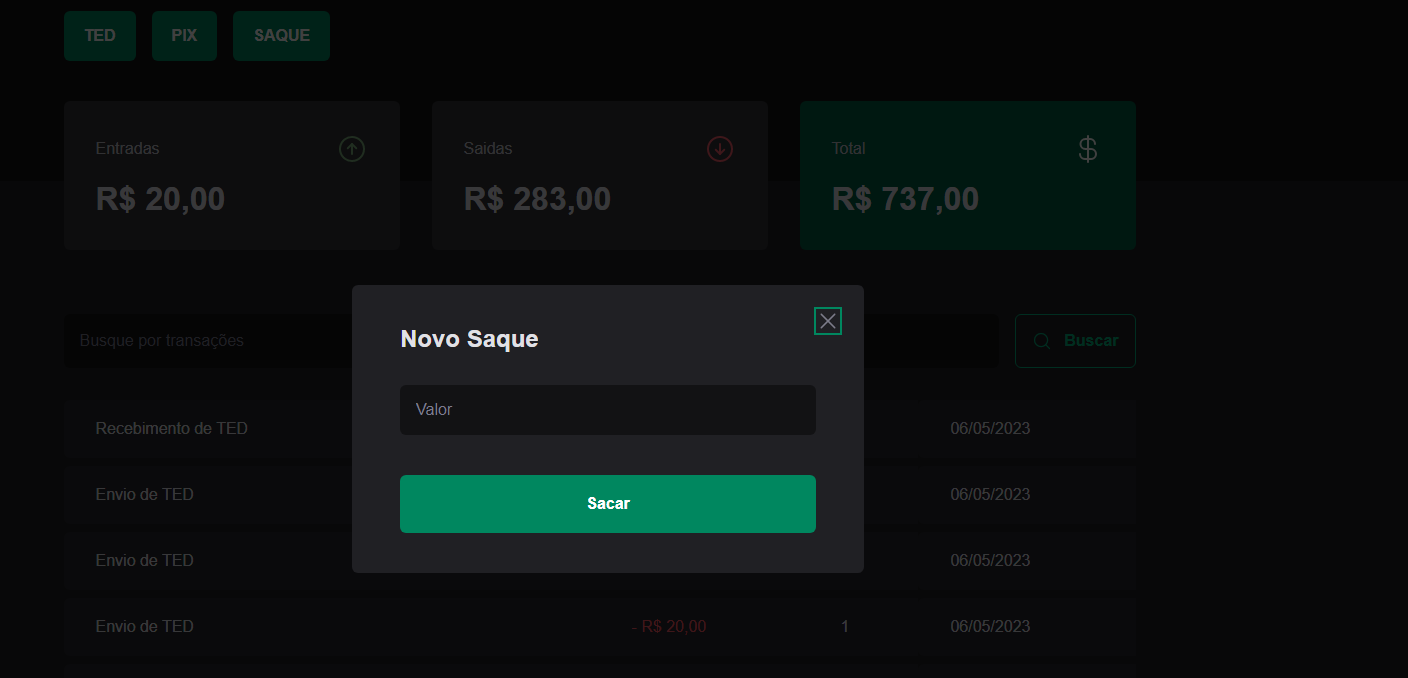
**Extrato -**



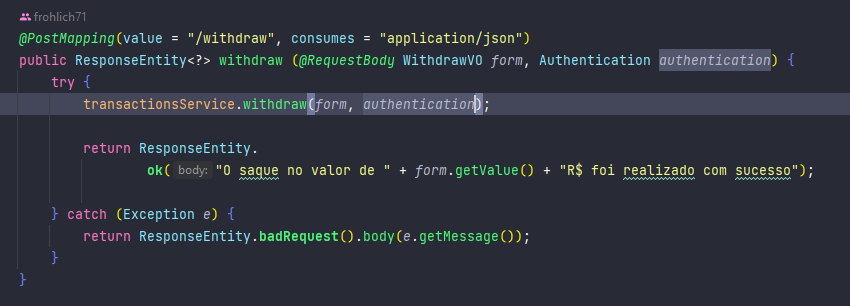
* Conforme sinalizado no dashboard de início, o extrato do cliente é gerado automaticamente de acordo com suas respectivas transferências, sendo permitido visualizar as últimas transações e também permitindo a pesquisa de uma transação em específico buscado pelo botão "Buscar";

**Saque -**

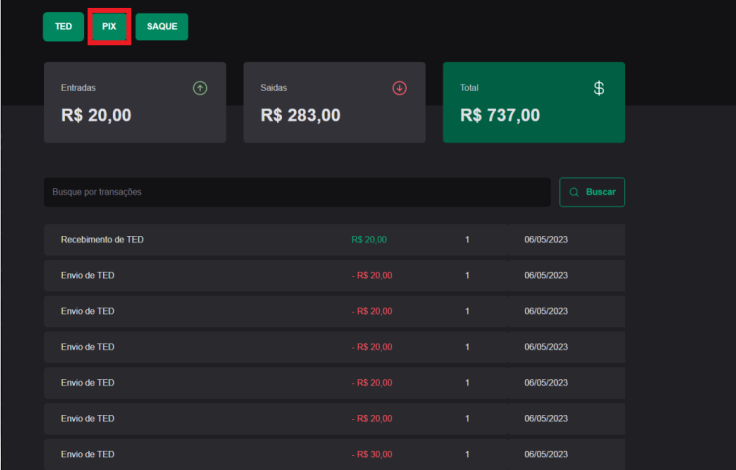
* Função disponibilizada para o cliente fazer a retirada de valores de sua conta;



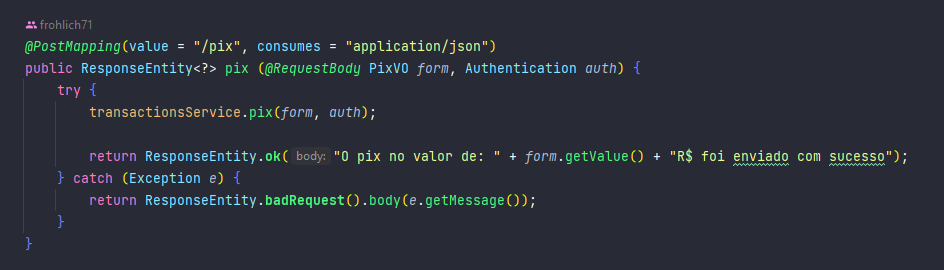
* + Ao clicar no botão de saque no dashboard, a classe principal chama o método de saque e subtrai o valor sobre o saldo atual do cliente:



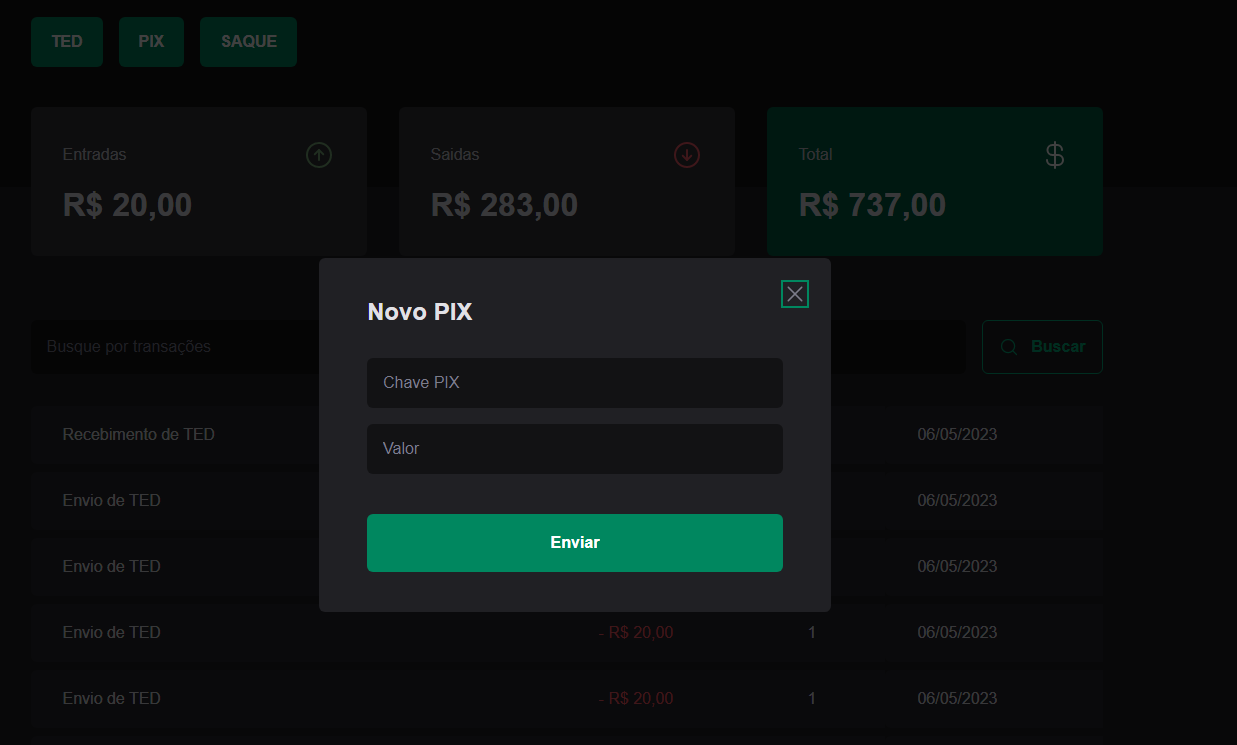
* **Transações:**

**PIX -**

* Selecionando o botão especificado no Dashboard como "PIX”, é permitido o usuário realizar uma transação PIX por meio da função construída no código fonte do projeto:



* Com a função PIX sendo chamada pelo arquivo principal, a função solicita a chave do destinatário e o valor a ser transacionado pelo usuário:



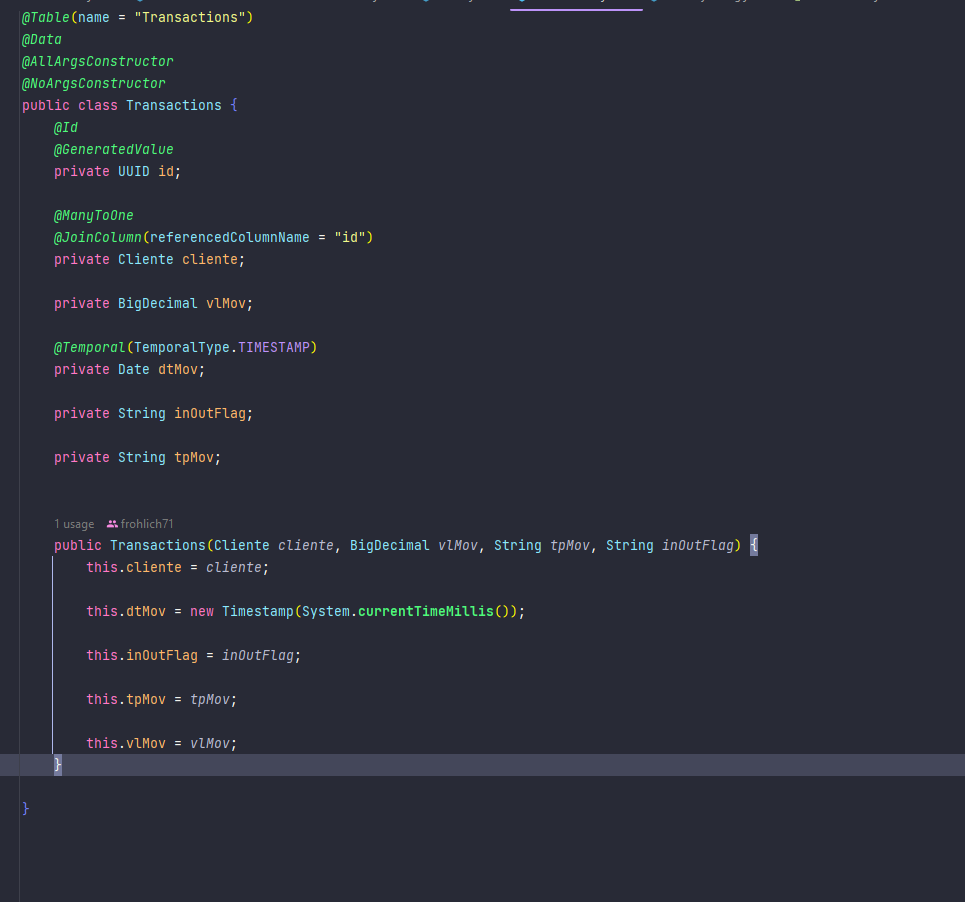
* Logo após o usuário realizar a transação PIX, seu saldo é atualizado e sua transação aparece logo abaixo no histórico de transações do dashboard;

**Cadastro PIX -**

* Método para cadastrar a chave PIX sendo chamado na classe PIX pela classe principal:



* Após o usuário chamar o método, um formulário é gerado para o usuário em JSON para a inserção dos dados de cadastro da sua chave PIX:

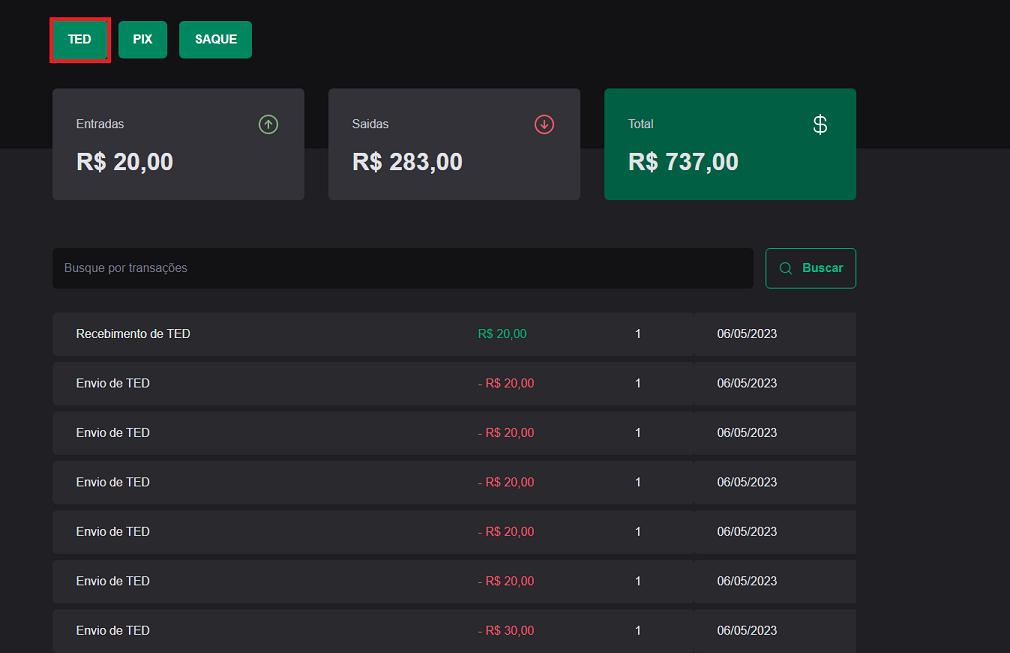


* **O método da criação do PIX chamado por meio de uma função na classe Transações que incrementa atributos herdados da classe cliente;**
* **Formulário gerado em JSON para criação da chave a ser implementada no método**



Sendo retornado o id com as informações da sua nova chave e a confirmação de que sua chave PIX foi registrada com sucesso no sistema;

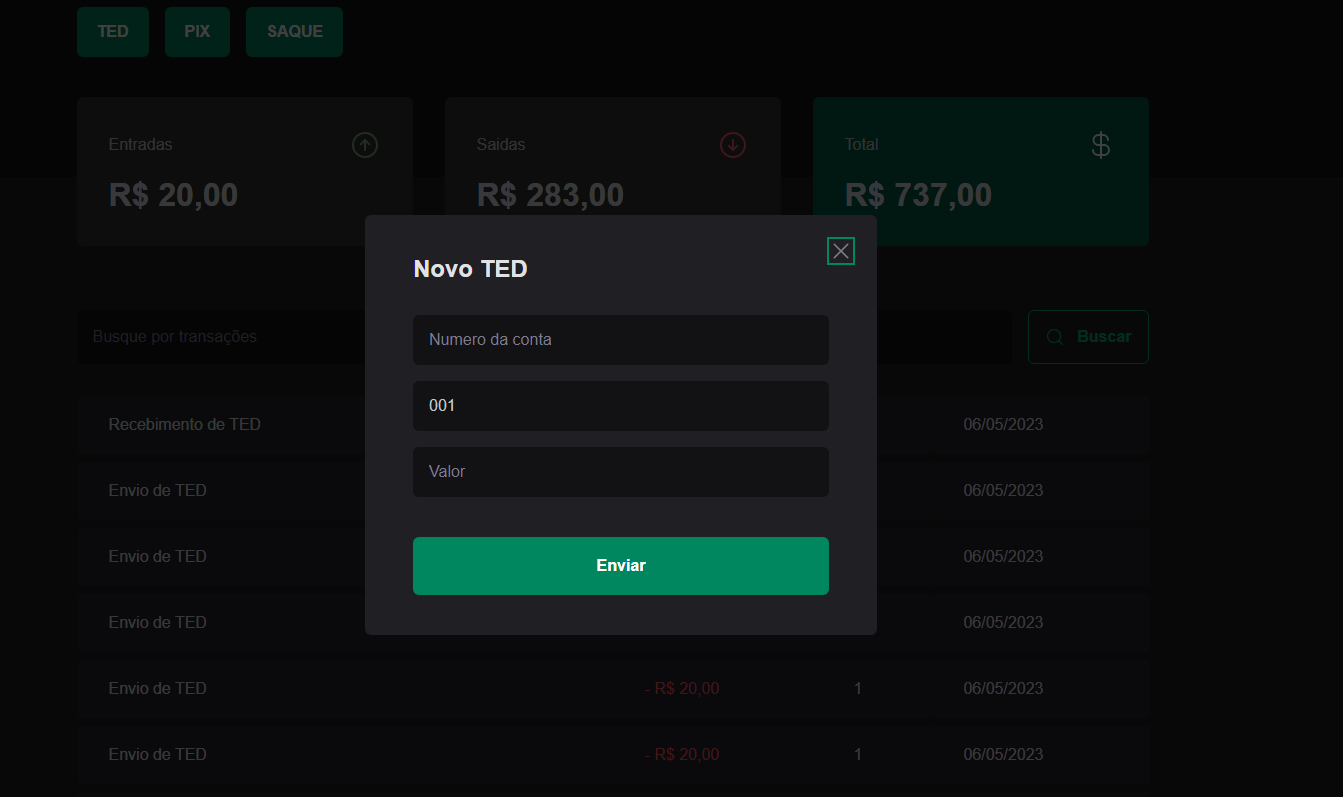
**TED -**



* Selecionando o botão especificado no Dashboard como "TED”, é permitido o usuário realizar uma transação TED por meio da função construída no código fonte do projeto:

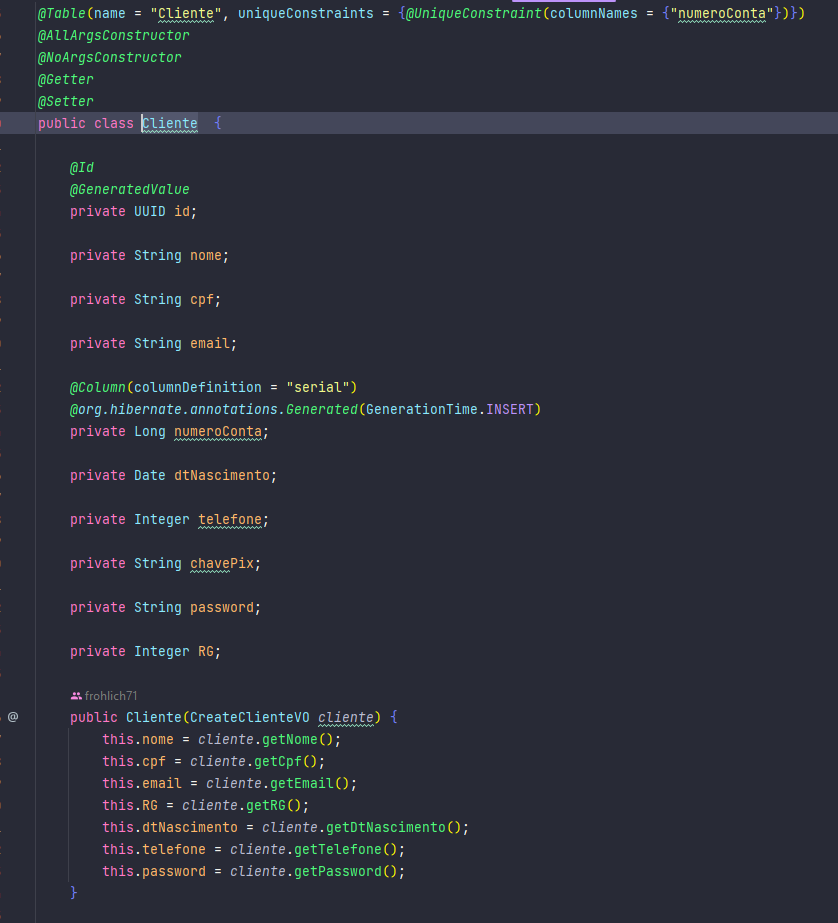


* Com a função TED sendo chamada pelo arquivo principal, a função solicita o número da conta e sua respectiva agência e também o valor a ser transacionado pelo usuário:

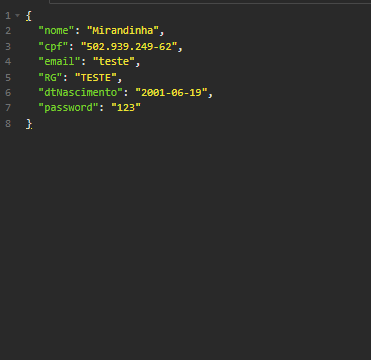


Logo após o usuário realizar o TED, seu saldo é atualizado e sua transação aparece logo abaixo no histórico de transações do dashboard;

**Cadastro de Clientes -**

* **CLASSE CLIENTE:** 
* **Classe Pai Cliente, é por meio desta classe que é gerado um formulário com os atributos do cliente para criação de um novo cliente em JSON;**
* **Método para cadastrar o cliente sendo chamado na classe Cliente pela classe principal:**



* **Após o usuário chamar o método, um formulário em JSON é gerado para o usuário para a inserção dos dados do novo cliente:**  
    
  
* **Sendo retornado o id com as informações do cliente e a confirmação de que o cliente foi inserido no sistema:**   
  